

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 63-085960

(43)Date of publication of application : 16.04.1988

(51)Int.Cl.

G06F 15/20

G06F 15/22

(21)Application number : 61-230148

(71)Applicant : CASIO COMPUT CO LTD

(22)Date of filing : 30.09.1986

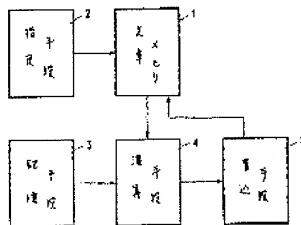
(72)Inventor : KUROSAWA HIROSHI

(54) TEXT PROCESSOR WITH TABLE CALCULATION FUNCTION

(57)Abstract:

PURPOSE: To efficiently perform the preparation of a table, by enabling vertical calculation, or horizontal calculation based on a numeric value array (two-dimensional numeric value array) in the table prepared on a text memory to be performed, in a word processor with table preparation function.

CONSTITUTION: An equation required for the vertical calculation or the horizontal calculation is inputted to a memory means 3. In this case, in the equation, the coordinate position data of a table item designated by a designating means 2 other than a various kinds of operators required for calculation are included. By storing a prescribed equation in the memory means 3 in such way, an arithmetic means 4 reads out numeric value data in the table item designated by the coordinate position data in the memory means 3, from the text memory 1, and executes a prescribed vertical calculation, or horizontal calculation, setting the numeric value data as a variable. An arithmetic result data obtained by the above operation is written in a prescribed item in the text memory 1. Assuming the same horizontal equation used for a first row is used for the rows behind a second row, by designating only the coordinate position in the table item which outputs an answer corresponding to each row, the horizontal equation behind the second row can be performed similarly by using the equation used in the first row as it is, that is, without changing the content of the calculation procedure memory 20.



LEGAL STATUS

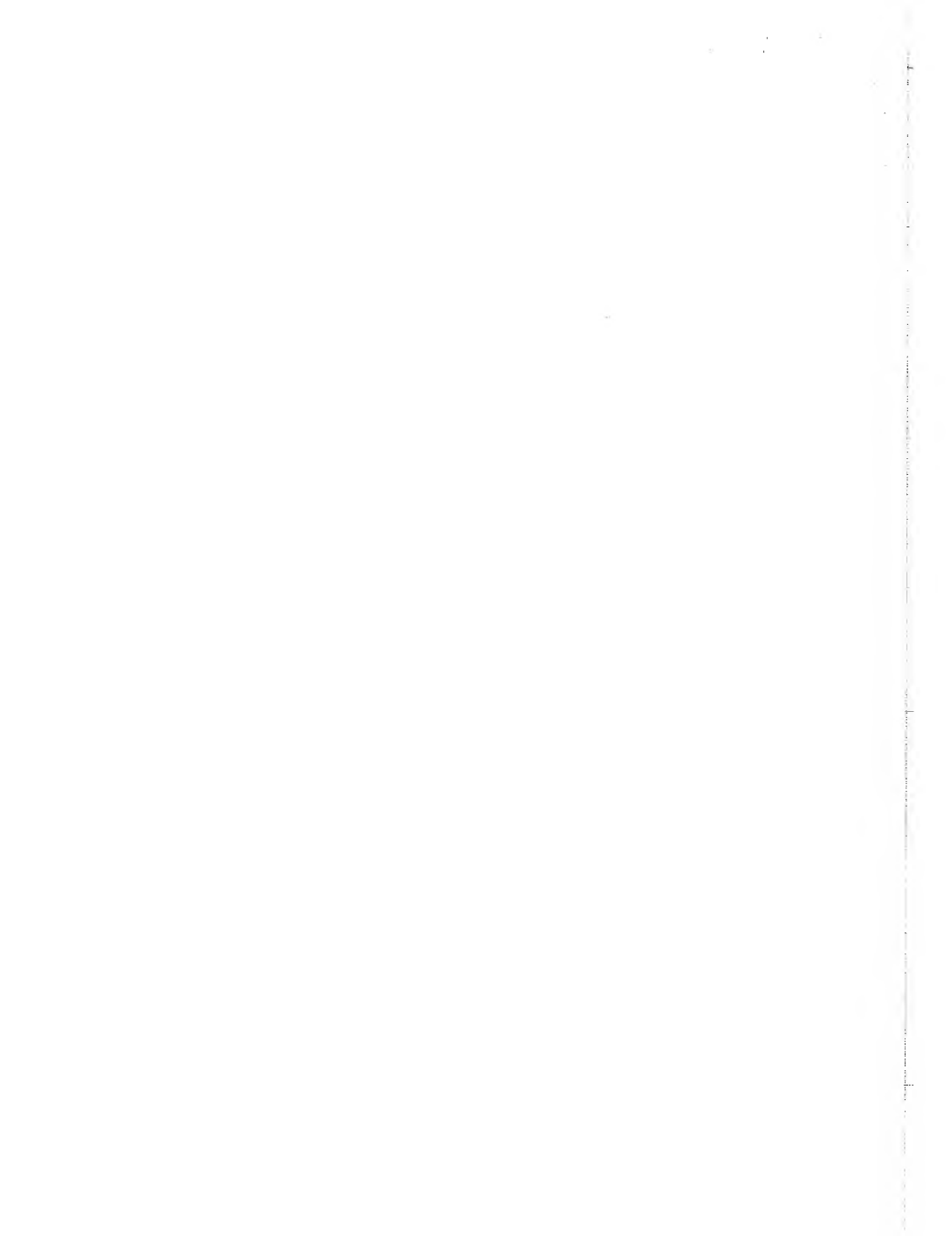
[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]



⑤ 日本国特許庁(JP)

⑥ 特許出願公開

⑦ 公開特許公報(A)

昭63-85960

⑧ Int. Cl.

G 06 F 15/20
15/22

識別記号

3 0 1

庁内整理番号

Y-7218-5B
7230-5B

⑨ 公開 昭和63年(1988)4月16日

審査請求 未請求 発明の数 1 (全8頁)

⑩ 発明の名称 表計算機能付き文章処理装置

⑪ 特 願 昭61-230148

⑫ 出 願 昭61(1986)9月30日

⑬ 発 明 者 黒 澤 宏 東京都西多摩郡羽村町栄町3丁目2番1号 カシオ計算機株式会社羽村技術センター内

⑭ 出 願 人 カシオ計算機株式会社 東京都新宿区西新宿2丁目6番1号

⑮ 代 理 人 弁理士 町田 俊正

明 細 書

表計算機能付き文章処理装置

1. 発明の名称

表計算機能付き文章処理装置

2. 特許請求の範囲

文章メモリ内に算線で囲まれた表を作成する作業機能を備えた文章処理装置において、

前記文章メモリ内に作成された表を構成する行数の表項目のうち、計算対象となる数値が記憶されている表項目を指定する指定手段と、この指定手段で指定された表項目に対応する前記文章メモリ上の座標位置データを含む計算式を記憶する記憶手段と、この記憶手段に記憶されている計算式内の座標位置データで指定された前記表項目内の数値データを前記文章メモリから読み出して前記記憶手段に記憶されている計算式の手順に従って計算を行う演算手段と、この演算手段で算出された計算結果を前記文章メモリ内の所定表項目内に書き込む書き込み手段とを具備したことを特徴とする発明

3. 発明の詳細な説明

【産業上の利用分野】

この発明は、ワードプロセッサ等において文章メモリ上に作成された表内の数値に基づいた表計算が実行可能な表計算機能付き文章処理装置に関する。

【発明の概要】

この発明は文章メモリ内に算線で囲まれた表を作成可能なワードプロセッサ等において、文章メモリ内に記憶されている表内の数値(変数)を予め記憶されている所定の計算式に基づいて計算することにより、乗除算等の表計算を文章メモリ内に作成された表に基づいて実行することができるようにしたものである。

【従来の技術】

従来、ワードプロセッサにおいては、任意の大

きさの表を文章内に作成することができる作表機能を備えたものが実用化され、これにより売上管理表や成績表あるいは、見込表等を容易に作成することができるようになっている。

【発明が解決しようとする問題点】

しかしながら、この種の作表機能付きワードプロセッサにおいて、例えば、売上管理表を作成するような場合に、商品別売上合計個数や合計金額あるいは総売上個数や総売上金額を予め小型電子式計算機等で計算しなければならず、その結果、作表を効率良く行うことができないという欠点があった。

この発明は上述した事情を背景になされたもので、その目的とするところは、文章メモリ上に作成された表内の数値配列（二次元数値配列）に基づいた縦計算や横計算が実行可能な作表機能付き文章処理装置を提供することにある。

く、この状態において、指定手段2は表内の各表項目のうち計算対象となる数値が記憶されている表項目を指定するが、この場合、例えばCRT画面に表示されている表を見ながらカーソルキーを操作して数値表示位置にカーソルをセットすることによって行なわれる。そして、記憶手段3へ縦計算や横計算に必要な計算式を入力する。この場合、計算式には必要とする各種の演算子の他、指定手段2によって指定された表項目の座標位置データが含まれる。このようにして記憶手段3に所定の計算式を記憶させておくと、演算手段4は、記憶手段3内の座標位置データで指定された表項目内の数値データを文章メモリ1から読み出してこれを変数として所定の縦計算や横計算を実行する。これによって得られた演算結果データは、文章メモリ1内の所定項目内に書き込まれる。

【実施例】

以下、この発明の一実施例を第2図～第8図に示す一実施例に基づいて具体的に説明する。な

【問題点を解決するための手段】

第1図はこの発明の機能ブロック図である。図中1は文章メモリ、2はこの文章メモリ1内に作成された表を構成する複数の表項目のうち、計算対象となる数値が記憶されている表項目を指定する指定手段、3はこの指定手段2で指定された表項目に対応する文章メモリ1上の座標位置データを含む計算式を記憶する記憶手段、4はこの記憶手段3に記憶されている計算式内の座標位置データで指定された前記表項目内の数値データを文章メモリ1から読み出して記憶手段3に記憶されている計算式の手順に従って計算を行う演算手段、5はこの演算手段4で算出された計算結果を文章メモリ1内の所定表項目内に書き込む直達手段である。

【作用】

この発明の作用について説明すると、文章メモリ1には野蠻で囲まれた表を作成すると共に、この表内に計算対象である数値を二次元配列してお

お、本実施例は作表機能付き日本語ワードプロセッサに適用した例を示している。

・構 造

第2図はこのワードプロセッサの基本的なブロック回路図である。図中11は、このワードプロセッサの入力部で、このキーボード上には、数字キーKA、文字キーKB、実行キーKC、ファンクションキー（四則演算キーを含む）KD、カーソルキーKEが設けられておくと共に、各種の表計算のうち表の横方向（折方向）に対する表計算を指定する横計算キーKF、縦方向（行方向）に対する表計算を指定する縦計算キーKG、表計算モードを解除する解除キーKHが設けられている。しかしして入力部11から操作キーに対応して出力されるキー入力信号は、入力制御部12に送られてキーコードに変換されたのちCPU（中央演算処理回路）13に取り込まれ、その入力処理プログラムを指定する。

CPU13は予め記憶されているマイクロプロ

グラムにしたがって入力処理、文章作成処理、表示処理、印字処理等を実行すると共に、本実施例にあっては表計算が実行可能となっている。そして、CPU13には表計算時における各種の計算モード（横計算モード、縦計算モード）を指定するモードレジスタ群が設けられている。また、CPU13にはテキストメモリ（文章メモリ）14が接続され、入力された文章データがコード化されて書き込まれる。

文章メモリ14はRAM（ランダムアクセスメモリ）によって構成され、CPU13の制御下でデータの書き込み、読み出し動作が制御されると共に、テキストポインタ15の値にしたがってその書き込みあるいは読み出しアドレスが指定される。

演算バッファ16は表計算実行時に計算手順メモリ20から読み出された計算手順が一次記憶されるもので、この内容は演算値17に送られてモードレジスタ群の内容に応じた表計算が実行される。この演算値17で算出された演算結果は、

B)を記憶するもので、これらは表計算実行用の式を入力する場合や表計算実行時に使用され、CPU13に取り込まれる。

動作

先ず、通常のキー操作手順にしたがって文章メモリ14内に界線で囲まれた表を作成する。例えば、第3図に示すような表を文章メモリ14内に作成したものとすると、この場合、表を構成する各表項目は横方向に3個、縦方向に3個設けられ、その下側3個、右側3個を除く各表項目には、上に「10」、「40」下に「20」、「30」の如く二次元配列された数値（計算対象）が書き込まれている。なお、図中、(1,1)～(3,3)は1行1桁目から3行3桁目までの表項目に対応し、また、文章メモリ14の座標位置は、図中左上端部を基点(1,1)として定められている。

次に、上述のようにして文章メモリ14内に作成された表に基いた表計算を実行する場合について

CPU13に取り込まれ、文章メモリ14に書き込まれる。

表示値18は例えばCRT表示装置によって構成され、CPU13から出力された表示用データが表示制御部19で表示駆動信号に変換されることにより1画面分のデータを表示する。

計算手順メモリ20は表計算実行用の計算手順を記憶するもので、計算手順として定義できるのは、横計算、縦計算である。また、出力位置メモリ21は、表計算の結果を表のどの位置に書き込むかの出力位置を記憶するもので、この出力位置は文章メモリ14の二次元座標によって表わされる。なお、計算手順メモリ20に書き込まれる計算手順や出力位置メモリ21に書き込まれる出力位置は入力値11から任意に入力設定されたもので、これらの内容は表計算実行時に、CPU13に取り込まれ、演算バッファ16にセットされる。

座標メモリ22は文章メモリ14の各種の座標位置データ(x₀, y₀)、(a, b)、(A、

で説明する。

第4図は横計算キーKcを操作したときに実行開始されるフローチャートである。文章メモリ14内に作成した表を表示値18に表示させた状態において、カーソルキーKxを操作してカーソルを移動し表の基準位置（第3図中左上端位置）にセットする。そして、実行キーKcを操作すると、その座標位置(4,2)が座標メモリ22に基準位置座標(x₀, y₀)として転送記憶される（ステップS1、S2）。

このようにして表の基準座標を入力したら、次に表計算を行うための式を入力する。すると、ステップS3ではカーソルキーKe、数字キーKa、ファンクションキーKdの操作に伴って入力された式を計算手順メモリ20に書き込む入力処理が実行される。

第5図はこの式入力処理の具体的な内容を説明する為のフローチャートである。先ず、ステップS3-1では式入力の為に操作されるキーの入力待ち状態となる。いま、文章メモリ14内の表項目

に記入されている数値を変数とする演算式

$$A + 2 \times B =$$

A: 表目(1, 1)内の数値

B: 表目(1, 2)内の数値

を入力するものとする。

先ず、数値変数Aの入力を指定する為に、カーソルキーBを操作して変数Aとなる数値の一部、例えば「10」の上位桁「1」にカーソルをセットする。いま、数値「10」の上位桁「1」にカーソルを合わせると、カーソル座標位置が更新される(ステップS3-5)。即ち、いま座標位置にセットされているカーソルを数値「10」の上位桁「1」にセットしたものとすると、座標メモリ22内の基準位置座標(x, y)の値(4, 2)に高づいてカーソル座標(a, b)が更新され、(6, 3)となる。そして、ファンクションキーKDを操作して演算子「+」を入力すると、ステップS3-2では、前回入力されたデータは数字であるかの判断が実行されるが、いま、最初の入力であるから、座標メモリ22内の

カーソル座標(a, b)が読み出されて計算手順メモリ20の先頭アドレス領域に書き込まれる(ステップS3-4)。そして、次のステップS3-3に進み、入力されたファンクションコードが計算手順メモリ20の次アドレス領域に書き込まれる。第7図はこの場合の計算手順メモリ20の記憶状態を示し、上述のように、カーソルを表内の数値「10」の一部にセットしたのち、ファンクションキーKDを操作して演算子「+」を入力すると、計算手順メモリ20の先頭アドレス領域には数値変数Aが直接入力されず、その座標位置(6, 3)が数値変数Aに代って書き込まれ、その後、次アドレス領域には入力された「+」のファンクションコードが書き込まれる。

そして、定数「2」を入力する為に、数字キーAを操作すると、ステップS3-6に進み、入力された数字コードが計算手順メモリ20の次アドレス領域に書き込まれる(第7図参照)。

続いてファンクションキーKDを操作して演算子「×」を入力する。この場合、ステップS3-

2では前回数字が入力されたかと判断されるので、入力されたファンクションコードが計算手順メモリ20の次アドレス領域に書き込まれる(第7図参照)。したがって、ファンクションキーKDが操作された場合に、前回入力されたデータが数字であれば、上述のようなカーソル座標(a, b)の書き込みは行なわれず、入力されたファンクションコード「×」がそのまま計算手順メモリ20に書き込まれる。

次に、カーソルキーKEを操作して変数Bとなる数値「40」の一部、例えば上位桁「4」にカーソルをセットする。すると、カーソル座標(a, b)は、(10, 3)となる(ステップS3-5)。その後、イコールコード「=」を入力すると、ステップS3-2からステップS3-4に進み、カーソル座標(10, 3)が計算手順メモリ20に書き込まれたのち、入力されたイコールコード「=」が計算手順メモリ20の次アドレスに書き込まれる(第7図参照)。

このようにして所望の式を入力し終ったら、実

行キーKCを操作する。すると、その式入力処理が終了し、第4図フローチャートの次ステップS4へ進行する。ここでは、演算が行なわれることによって得られた演算結果データを表のどの位置(文なメモリ14のどの位置)に記憶させるかを指定する為にカーソルキーKEを操作してその指定位置へカーソルを移動させると、このときのカーソル座標が出力位置メモリ21へ転送されて記憶される。また、このときのカーソル座標は、座標メモリ22に演算結果の出力位置座標(A, B)として転送される(ステップS5)。第8図はこのようにして入力された座標データが出力位置メモリ21に記憶された状態を示している。

しかし、次のステップS8では演算を実行開始する為に、先ず、計算手順メモリ20の先頭アドレス領域内のデータが読み出される。ここで、第7図の例では、計算手順メモリ20から座標データ(6, 3)が読み出される。しかし、ステップS7に進み、計算手順メモリ20から読み出したデータに応じた演算処理が実行されたの

ち、ステップS8に戻り、計算手順メモリ20に設定されている式の演算処理が全て終るまで上述の動作が繰り返される。

第6図は、第4図で示した演算処理(ステップS7)の具体的な内容を示したフローチャートである。このフローに入ると、計算手順メモリ20から読み出したデータに応じた演算処理に進むが、いま、第7図の例では計算手順メモリ20から座標データ(6, 3)が読み出されるので、ステップS7-1に進み、この座標データを座標メモリ22内の座標(a, b)としてセットする。そして、カーソルが座標(a, b)位置へ目的的に移動される(ステップS7-2)。この場合、X座標aは計算手順メモリ20から読み出されたX座標(6)、Y座標bは出力位置のY座標(3)であるから座標(a, b)は(6, 3)となり、この場合においては、計算手順メモリ20から読み出された座標と同様となる。その後、カーソル座標(a, b)を含む位置にある表項目内のデータが全て読み出されて演算バッファ16の先頭位置

から順次書き込まれる(ステップS7-3)。ここで、該当表項目から全てのデータを読み出す場合には、次の如くして行なわれる。例えば、第3図に示すように文章メモリ14の表内において、カーソル座標(6, 3)を含む位置にある表項目内の数値変数「10」を取り出す場合、カーソル座標(6, 3)から左方向の昇線コードがサーチされるまでテキストポイント15のX座標を「1」ずつ減算し、そして、昇線コードが検索されると、今度はテキストポイント15のX座標を「1」ずつ加算してX座標を右方向へ移動させてゆきデータの先頭を見つける。そして、データの先頭が検し出されると、その先頭から1桁ずつ右方向へ移動しながらデータを順次取り出してゆき、次の昇線コードが来るまで右方向への移動を続ける。これによって1項目内に記憶されている全てのデータをその先頭から順番に取り出すことができる。その結果、上述の式入力処理時において、表項目内のデータ記憶領域、つまりその先頭から最後の位置までを特定しなくとも、その

データ内の一部を指定しておくだけで、表計算処理にその表項目内のデータ全てを取り出すことができる。このようにして文章メモリ14から取り出された1表項目内の数値変数は、演算バッファ18に転送される。

そして、第4図のステップS8に戻り、出力位置メモリ21から次アドレス領域内のデータが読み出されるが、第7図の例ではファンクションコード「+」が読み出されるので、第6図のフローにおいて、次にステップS7-4に進み、計算手順メモリ20から読み出したファンクションコードが演算バッファ18に取り込まれる。また、次に計算手順メモリ20から数字コードが読み出されるが、この場合も、ステップS7-4が実行され、数字コードがそのまま演算バッファ18に取り込まれる。そして、次に、計算手順メモリ20からファンクションコード「×」が読み出されて演算バッファ18に取り込まれる。

そして、次のタイミングでは計算手順メモリ20から座標データ(10, 3)が読み出され、

ステップS7-1～S7-3の実行に移る。その結果、上述と同様に、文章メモリ14の該当表項目から全ての数値変数「40」が読み出されて演算バッファ18に取り込まれる。

最後に、計算手順メモリ20から「=」コードが読み出されると、ステップS7-5に進み、演算バッファ18の内容に基づいた演算が実行される。即ち、

$$10 + 2 \times 40 = 90$$

の計算が行なわれる。そして、この演算結果は、座標メモリ22から座標(A, B)を読み出して文章メモリ14の対応表項目内に書き込まれる。即ち、座標(A, B)は、いまの場合(13, 3)で、この座標を含む表項目(1, 3)内に、その座標位置から順に演算結果「90」が書き込まれる。

このように「=」コードの検出に伴って計算手順メモリ20に予め設定した式に応じた演算が実行され、その答が文章メモリ14の指定位置に書き込まれるが、その後、計算手順メモリ20の

次アドレス領域にはコードが記憶されていないので、次に、第4図ステップS3に戻り、次の行に対して横計算を行うための式入力可能状態となる。

いま、第3図に示す2行目の横計算を行う場合、その計算式が、数値変数を除き第1行目のそれと変わりが無いときには、その横計算結果を出力する表の出力位置だけを指定すると、第2行目の横計算についても第1行目の式(第7図)が自動的に選択され、その結果(答)が指定出力位置に書き込まれるようになる。

即ち、カーソルキーKEを操作して第2行目の横計算で得られた計算結果を出力する表の出力位置へカーソルを移動させる。いま、表項目(2, 3)内の座標位置(13, 5)にカーソルをセットすると、それに応じて第5図ステップS3-5ではカーソル座標(a, b)が更新されて(13, 5)となる。そして、実行キーKCを操作すると、ステップS4に進み、カーソル座標(13, 5)が出力位置メモリ21に転送記憶される(第8図参照)。また、カーソル座標

(13, 5)は座標メモリ22に計算結果の出力位置座標(A, B)として転送される(ステップS5)。そして、ステップS6, S7に進み、第2行目に対しての横計算が実行される。この場合、先ず、計算手順メモリ20から座標データ(6, 3)が読み出される(ステップS7-1)。そして、次のステップS7-2ではカーソル座標(a, b)の位置に移動される。ここで、X座標aは計算手順メモリ20から読み出されたX座標(6)、Y座標bは今回指定された出力位置のY座標(5)であるから表項目(2, 1)の座標位置(6, 5)にカーソルが移動される。これによって、次のステップS7-3でこの座標を含む項目データ「20」が読み出されて演算バッファ16に転送される。以下、上述の第1行目の横計算と同様に、計算手順メモリ20から演算子「+」、定数「2」、演算子「×」が順次読み出されて演算バッファ16に転送される。そして次に、計算手順メモリ20から座標データ(10, 3)が読み出されてこの値が(10, 5)に変換

され(ステップS7-1, S7-2)、これによって指定された表項目(2, 2)から項目データ(30)が読み出されて演算バッファ16に転送される。そして、「=」のコード検出に伴って演算バッファ16にセットされた式に応じた計算処理が実行される(ステップS7-5)。即ち、

$$20 + 2 \times 30 = 80$$

の計算が行われ、この計算結果が座標(13, 5)に書き込まれる。

このように、第1行目の横計算式が表項目(1, 1) + 2 × 表項目(1, 2) → 表項目(1, 4)である場合において、2行目以降も同様の式を用いた横計算を行うものとする。各行に対応してその答を出力させる表項目内の座標位置を指定するだけで、第1行目で採用した式をそのまま使用して、即ち、計算手順メモリ20の内容を変更せずにそのまま使用して第2行目以降の横計算も同様に行うことができる。したがって、表項目(2, 4)、(3, 4)、(5, 4)内の座標を指定するだけで、

表項目(2, 1) + 2 × 表項目(2, 2) → 表項目(2, 4)

表項目(3, 1) + 2 × 表項目(3, 2) → 表項目(3, 4)

表項目(5, 1) + 2 × 表項目(5, 2) → 表項目(5, 4)

の計算が実行される。

このように第1行目で採用した式をそのまま第2行目以降にも連続的に使用することができるので、同一式の計算が行方向に連続する横計算においては、式入力操作の簡略化を図ることができると共に、メモリの節約ともなり、有効である。また、上述の例のように、第4行を飛ばして第5行目を指定することもでき、指定データの連続性には依存しない。

しかし、解除キーKJが操作されると、ステップS3でそのことが検出されてこの横計算フローから抜け、横計算モードが解除される。

一方、縦計算も横計算と同様に実行されるので、その説明を省略するが、縦計算については横

計算の場合に比べて単に相定方向が上下と左右との違いだけであり、また、縦計算についても横計算と同様に四則演算子や定数を任意に入力することにより所望の式に基づいた計算を実行させることができる。

なお、上記実施例は、四則演算について説明したが、関数計算等であってもよい。

【発明の効果】

この発明は以上詳述に説明したように、文章メモリ内に昇順で固まれた表を作成可能なワードプロセッサ等において、文章メモリ内に記憶されている表内の数値配列を予め記憶されている所定の計算式に基づいて計算するようにしたから、乗積集計等の表計算を文章メモリ内に作成された表に基づいて計算することができ、極めて便利な実用性の高いものとなる。

4. 図面の簡単な説明

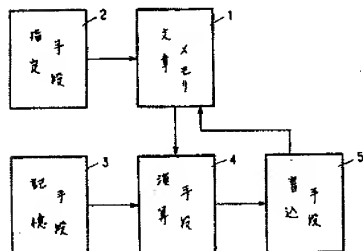
第1図はこの発明の機能ブロック図、第2図～

第8図はこの発明の実施例を示し、第2図はこの発明が適用した日本語ワードプロセッサの基本的なブロック回路図、第3図は第2図で示した文章メモリ14内に作成された表を示した図、第4図は横計算を行う場合のフローチャート、第5図は第4図で示した式入力処理の具体的な内容を説明する為のフローチャート、第6図は第4図で示した演算処理の具体的な内容を説明する為のフローチャート、第7図は計算手順メモリ20の記憶状態図、第8図は出力位置メモリ21の記憶状態図である。

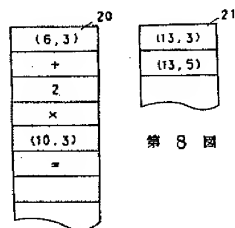
11……入力部、13……CPU、14……文章メモリ、18……演算バッファ、17……演算部、20……計算手順メモリ、21……出力位置メモリ。

特許出願人 カシオ計算機株式会社

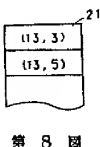
代理人 弁理士 町田 義 正



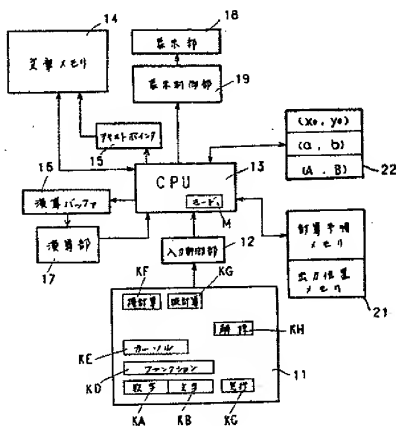
第1図



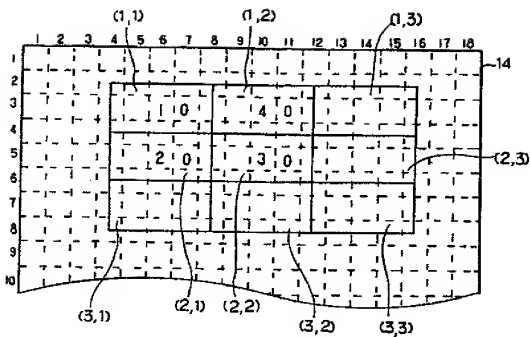
第7図



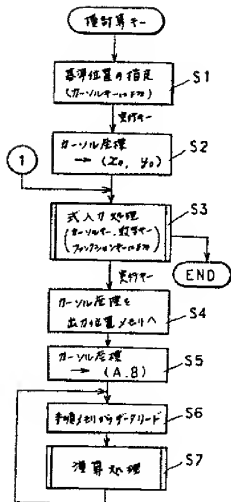
第8図



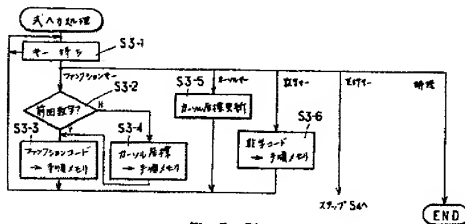
第2図



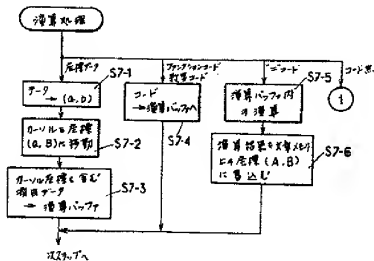
第 3 図



第 4 図



第 5 図



第 6 図